

AMENDMENT TO THE CLAIMS**Claims pending**

- At time of the Action: Claims 1-43.
- After this Response: Claims 1-41.

Canceled or Withdrawn claims: 42-43

Amended claims: None

New claims: None

1. **(Original)** A software architecture for a distributed computing system comprising:

an application configured to handle requests submitted by remote devices over a network; and

an application program interface to present functions used by the application to access network and computing resources of the distributed computing system.

2. **(Original)** A software architecture as recited in claim 1, wherein the distributed computing system comprises client devices and server devices that handle requests from the client devices, the remote devices comprising at least one client device.

3. **(Original)** A software architecture as recited in claim 1, wherein the distributed computing system comprises client devices and server devices that handle requests from the client devices, the remote devices comprising at least one server device that is configured as a Web server.

1
2 4. (Original) A software architecture as recited in claim 1, wherein the
3 application program interface comprises:

- 4 a first group of services related to creating Web applications;
5 a second group of services related to constructing client applications;
6 a third group of services related to data and handling XML documents; and
7 a fourth group of services related to base class libraries.

8
9 5. (Original) An application program interface embodied on one or
10 more computer readable media, comprising:

- 11 a first group of services related to creating Web applications;
12 a second group of services related to constructing client applications;
13 a third group of services related to data and handling XML documents; and
14 a fourth group of services related to base class libraries.

15
16 6. (Original) An application program interface as recited in claim 5,
17 wherein the first group of services comprises:

- 18 first functions that enable construction and use of Web services;
19 second functions that enable temporary caching of frequently used
20 resources;
21 third functions that enable initial configuration;
22 fourth functions that enable creation of controls and Web pages;
23 fifth functions that enable security in Web server applications; and
24 sixth functions that enable access to session state values.
25

1 7. **(Original)** An application program interface as recited in claim 5,
2 wherein the second group of services comprises:

3 first functions that enable creation of windowing graphical user interface
4 environments; and

5 second functions that enable graphical functionality.

6
7 8. **(Original)** An application program interface as recited in claim 5,
8 wherein the third group of services comprises:

9 first functions that enable management of data from multiple data sources;
10 and

11 second functions that enable XML processing.

12
13 9. **(Original)** An application program interface as recited in claim 5,
14 wherein the fourth group of services comprises:

15 first functions that enable definitions of various collections of objects;

16 second functions that enable programmatic access to configuration settings
17 and handling of errors in configuration files;

18 third functions that enable application debugging and code execution
19 tracing;

20 fourth functions that enable customization of data according to cultural
21 related information;

22 fifth functions that enable input/output of data;

23 sixth functions that enable a programming interface to network protocols;

24 seventh functions that enable a managed view of types, methods, and fields;

1 eighth functions that enable creation, storage and management of various
2 culture-specific resources;

3 ninth functions that enable system security and permissions;

4 tenth functions that enable installation and running of services;

5 eleventh functions that enable character encoding;

6 twelfth functions that enable multi-threaded programming; and

7 thirteenth functions that facilitate runtime operations.

8
9 10. (Original) A network software architecture comprising the
10 application program interface as recited in claim 5.

11
12 11. (Original) A distributed computer software architecture,
13 comprising:

14 one or more applications configured to be executed on one or more
15 computing devices, the applications handling requests submitted from remote
16 computing devices;

17 a networking platform to support the one or more applications; and

18 an application programming interface to interface the one or more
19 applications with the networking platform.

20
21 12. (Original) A distributed computer software architecture as recited in
22 claim 11, further comprising a remote application configured to be executed on
23 one of the remote computing devices, the remote application using the application
24 programming interface to access the networking platform.

25

1 13. (Original) A distributed computer software architecture as recited in
2 claim 11, wherein the application programming interface comprises:

- 3 a first group of services related to creating Web applications;
4 a second group of services related to constructing client applications;
5 a third group of services related to data and handling XML documents; and
6 a fourth group of services related to base class libraries.

7
8 14. (Original) A distributed computer software architecture as recited in
9 claim 11, wherein the application programming interface exposes multiple
10 functions comprising:

- 11 first functions that enable construction and use of Web services;
12 second functions that enable temporary caching of frequently used
13 resources;
14 third functions that enable initial configuration;
15 fourth functions that enable creation of controls and Web pages;
16 fifth functions that enable security in Web server applications; and
17 sixth functions that enable access to session state values.

18
19 15. (Original) A distributed computer software architecture as recited in
20 claim 11, wherein the application programming interface exposes multiple
21 functions comprising:

- 22 first functions that enable creation of windowing graphical user interface
23 environments; and
24 second functions that enable graphical functionality.
25

1 16. (Original) A distributed computer software architecture as recited in
2 claim 11, wherein the application programming interface exposes multiple
3 functions comprising:

4 first functions that enable management of data from multiple data sources;
5 and
6 second functions that enable XML processing.

7
8 17. (Original) A distributed computer software architecture as recited in
9 claim 11, wherein the application programming interface exposes multiple
10 functions comprising:

11 first functions that enable definitions of various collections of objects;
12 second functions that enable programmatic access to configuration settings
13 and handling of errors in configuration files;
14 third functions that enable application debugging and code execution
15 tracing;
16 fourth functions that enable customization of data according to cultural
17 related information;
18 fifth functions that enable input/output of data;
19 sixth functions that enable a programming interface to network protocols;
20 seventh functions that enable a managed view of loaded types, methods,
21 and fields;
22 eighth functions that enable creation, storage and management of various
23 culture-specific resources;
24 ninth functions that enable system security and permissions;
25 tenth functions that enable installation and running of services;

1 eleventh functions that enable character encoding;
2 twelfth functions that enable multi-threaded programming; and
3 thirteenth functions that facilitate runtime operations.
4

5 18. (Original) A computer system including one or more
6 microprocessors and one or more software programs, the one or more software
7 programs utilizing an application program interface to request services from an
8 operating system, the application program interface including separate commands
9 to request services consisting of the following groups of services:

10 A. a first group of services related to creating Web applications:

11 constructing Web services;
12 temporary caching resources;
13 performing initial configuration;
14 creating controls and Web pages;
15 enabling security in Web server applications;
16 accessing session state values;

17 B. a second group of services related to constructing client applications:

18 creating windowing graphical user interface environments;
19 enabling graphical functionality;

20 C. a third group of services related to data and handling XML documents:

21 enabling management of data from multiple data sources;
22 second functions that enable XML processing.

23 D. a fourth group of services related to base class libraries:

24 defining various collections of objects;
25

1 accessing configuration settings and handling errors in configuration
2 files;
3 debugging and tracing code execution;
4 customizing data according to cultural related information;
5 inputting and outputting of data;
6 enabling a programming interface to network protocols;
7 viewing loaded types, methods, and fields;
8 creating, storing and managing various culture-specific resources;
9 enabling system security and permissions;
10 installing and running services;
11 enabling character encoding;
12 enabling multi-threaded programming; and
13 facilitating runtime operations.

14
15 19. (Original) A system comprising:
16 means for exposing a first set of functions that enable browser/server
17 communication;
18 means for exposing a second set of functions that enable drawing and
19 construction of client applications;
20 means for exposing a third set of functions that enable connectivity to data
21 sources and XML functionality; and
22 means for exposing a fourth set of functions that enable system and runtime
23 functionality.
24
25

1 20. **(Original)** A system as recited in claim 19, wherein the first set of
2 functions comprises:

3 first functions that enable construction and use of Web services;
4 second functions that enable temporary caching of frequently used
5 resources;
6 third functions that enable initial configuration;
7 fourth functions that enable creation of controls and Web pages;
8 fifth functions that enable security in Web server applications; and
9 sixth functions that enable access to session state values.

10
11 21. **(Original)** A system as recited in claim 19, wherein the second set
12 of functions comprises:

13 first functions that enable creation of windowing graphical user interface
14 environments; and
15 second functions that enable graphical functionality.

16
17 22. **(Original)** A system as recited in claim 19, wherein the third set of
18 functions comprises:

19 first functions that enable management of data from multiple data sources;
20 and
21 second functions that enable XML processing.

22
23 23. **(Original)** A system as recited in claim 19, wherein the fourth set of
24 functions comprises:

25 first functions that enable definitions of various collections of objects;

1 second functions that enable programmatic access to configuration settings
2 and handling of errors in configuration files;

3 third functions that enable application debugging and code execution
4 tracing;

5 fourth functions that enable customization of data according to cultural
6 related information;

7 fifth functions that enable input/output of data;

8 sixth functions that enable a programming interface to network protocols;

9 seventh functions that enable a managed view of loaded types, methods,
10 and fields;

11 eighth functions that enable creation, storage and management of various
12 culture-specific resources;

13 ninth functions that enable system security and permissions;

14 tenth functions that enable installation and running of services;

15 eleventh functions that enable character encoding;

16 twelfth functions that enable multi-threaded programming; and

17 thirteenth functions that facilitate runtime operations.

18
19 24. (Original) A method, comprising:

20 managing network and computing resources for a distributed computing
21 system; and

22 exposing a set of functions that enable developers to access the network and
23 computing resources of the distributed computing system, the set of functions
24 comprising first functions to facilitate browser/server communication, second
25 functions to facilitate construction of client applications, third functions to

1 facilitate connectivity to data sources and XML functionality, and fourth functions
2 to access system and runtime resources.

3
4 25. (Original) A method as recited in claim 24, further comprising
5 receiving a request from a remote computing device, the request containing a call
6 to at least one of the first, second, third, and fourth functions.

7
8 26. (Original) A method, comprising:
9 creating a first namespace with functions that enable browser/server
10 communication;
11 creating a second namespace with functions that enable drawing and
12 construction of client applications;
13 creating a third namespace with functions that enable connectivity to data
14 sources and XML functionality; and
15 creating a fourth namespace with functions that enable system and runtime
16 functionality.

17
18 27. (Original) A method as recited in claim 26, wherein the first
19 namespace defines classes that facilitate:
20 construction and use of Web services;
21 temporary caching of resources;
22 initial configuration;
23 creation of controls and Web pages;
24 security in Web server applications; and
25 access to session state values.

1
2 28. (Original) A method as recited in claim 26, wherein the second
3 namespace defines classes that facilitate:
4 creation of windowing graphical user interface environments; and
5 graphical functionality.
6

7 29. (Original) A method as recited in claim 26, wherein the third
8 namespace defines classes that facilitate:
9 management of data from multiple data sources; and
10 processing of XML documents.
11

12 30. (Original) A method as recited in claim 26, wherein the fourth
13 namespace defines classes that facilitate:
14 programmatic access to configuration settings and handling of errors in
15 configuration files;
16 application debugging and code execution tracing;
17 customization of data according to cultural related information;
18 inputting and outputting of data;
19 interfacing to network protocols;
20 viewing loaded types, methods, and fields;
21 creation, storage and management of various culture-specific resources;
22 system security and permissions;
23 installation and running of services;
24 character encoding;
25 multi-threaded programming; and

runtime operations.

31. (Original) A method, comprising:

calling one or more first functions to facilitate browser/server communication;

calling one or more second functions to facilitate construction of client applications;

calling one or more third functions to facilitate connectivity to data sources and XML functionality; and

calling one or more fourth functions to access system and runtime resources.

32. (Original) A method as recited in claim 36, wherein the first functions comprise functions for construction and use of Web services, temporary caching of resources, initial configuration, creation of controls and pages that will appear as user interfaces, securing Web server applications, and accessing session state values.

33. (Original) A method as recited in claim 36, wherein the second functions comprise functions for creation of windowing graphical user interface environments, and graphical functionality.

34. (Original) A method as recited in claim 36, wherein the third functions comprise functions for management of data from multiple data sources, and XML processing.

1
2 35. (Original) A method as recited in claim 36, wherein the fourth
3 functions comprise functions for programmatic access to configuration settings,
4 application debugging and code execution tracing, customization of text according
5 to cultural related information, synchronous and asynchronous reading from and
6 writing to data streams and files, creation and management of various culture-
7 specific resources, system security and permissions, installation and running of
8 services, character encoding, and multi-threaded programming.

9
10 36. (Original) A method, comprising:
11 receiving one or more calls to one or more first functions to facilitate
12 browser/server communication;
13 receiving one or more calls to one or more second functions to facilitate
14 construction of client applications;
15 receiving one or more calls to one or more third functions to facilitate
16 connectivity to data sources and XML functionality; and
17 receiving one or more calls to one or more fourth functions to access
18 system and runtime resources.

19
20 37. (Original) A method as recited in claim 31, wherein the first
21 functions comprise functions for construction and use of Web services, temporary
22 caching of resources, initial configuration, creation of controls and pages that will
23 appear as user interfaces, securing Web server applications, and accessing session
24 state values.
25

1 38. (Original) A method as recited in claim 31, wherein the second
2 functions comprise functions for creation of windowing graphical user interface
3 environments, and graphical functionality.

4
5 39. (Original) A method as recited in claim 31, wherein the third
6 functions comprise functions for management of data from multiple data sources,
7 and XML processing.

8
9 40. (Original) A method as recited in claim 31, wherein the fourth
10 functions comprise functions for programmatic access to configuration settings,
11 application debugging and code execution tracing, customization of text according
12 to cultural related information, synchronous and asynchronous reading from and
13 writing to data streams and files, creation and management of various culture-
14 specific resources, system security and permissions, installation and running of
15 services, character encoding, and multi-threaded programming.

16
17 41. (Original) A method for exposing resources using an application
18 program interface, comprising:

19 A. exposing a first group of services related to creating Web applications,
20 including:

21 constructing Web services;
22 temporary caching resources;
23 performing initial configuration;
24 creating controls and Web pages;
25 enabling security in Web server applications;

1 accessing session state values;

2 B. exposing a second group of services related to constructing client
3 applications, including:

4 creating windowing graphical user interface environments;

5 enabling graphical functionality;

6 C. exposing a third group of services related to data and handling XML
7 documents, including:

8 enabling management of data from multiple data sources;

9 second functions that enable XML processing.

10 D. exposing a fourth group of services related to base class libraries,
11 including:

12 defining various collections of objects;

13 accessing configuration settings and handling errors in configuration
14 files;

15 debugging and tracing code execution;

16 customizing data according to cultural related information;

17 inputting and outputting of data;

18 enabling a programming interface to network protocols;

19 viewing loaded types, methods, and fields;

20 creating, storing and managing various culture-specific resources;

21 enabling system security and permissions;

22 installing and running services;

23 enabling character encoding;

24 enabling multi-threaded programming; and

25 facilitating runtime operations.

1
2 42. (Canceled) A method of organizing a set of types into a hierarchical
3 namespace comprising:

4 creating a plurality of groups from the set of types, each group containing at
5 least one type that exposes logically related functionality;

6 assigning a name to each group in the plurality; and

7 selecting a top level identifier and prefixing the name of each group with
8 the top level identifier so that the types in each group are referenced by a
9 hierarchical name that includes the selected top level identifier prefixed to the
10 name of the group containing the type.

11
12 43. (Canceled) A system comprising:

13 a set of types, each type comprising one of a delegate, an enumeration, an
14 interface, a class, and a structure; and

15 a namespace defined by the set of types to provide access to logically
16 related functionality of a computing system.